What is claimed is:

1. A method of monitoring an embedded system, the method comprising:

identifying programs to be monitored;

specifying a delta time within which each identified program will check-in;

specifying a remedial action to be taken in the event the identified program fails to check-in within the delta time;

for each identified program periodically determining whether the time of the last check-in is greater than a current time minus the delta time; and

when the time of the last check in for any identified program is less than a current time minus the delta time for that program executing the remedial action associated with that program.

2. A method, as set forth in claim 1, further comprising:

identifying a second and third remedial action;

the second time the last check in for any identified program is less than a current time minus the delta time for that program, executing the second remedial action associated with that program; and

the third time the last check in for any identified program is less than a current time minus the delta time for that program to be monitored executing the third remedial action associated with that program.

3. A method, as set forth in claim 1, wherein the remedial action comprises restarting the system.

4. A method, as set forth in claim 2, wherein first remedial action comprises restarting he identified program.

5. A method, as set forth in claim 4, wherein the second remedial action comprises restarting the system.

6. A method, as set forth in claim 4, wherein the third remedial action comprises halting the system.

7. A method, as set forth in claim 6, wherein the third remedial action further comprises indicating that the system is no longer functional.

8. A method as set forth in claim 7, wherein indicating the system is no longer function comprises illuminating an indicator.

9. A method as set forth in claim 1, wherein the remedial action comprises illuminating an indicator to indicate that the system is not functioning correctly.

10. A method, as set forth in claim 1, further comprising:

registering each identified program by creating an entry containing an identifier of the identified program, the delta time for the identified program and an indication of the first remedial action.

11. A method, as set forth in claim 10, wherein the entry is a key in a registry associated with an operating system of the embedded system.

12. A method, as set forth in claim 1, wherein the step of specifying a remedial action comprises:

for each identified program creating a list of executable files which, when executed perform remedial actions; and

creating a pointer, for each identified program, into each list which may be modified to point to individual entries in the list.

13. A method, as set forth in claim 1, wherein the remedial action comprises rebooting the system and wherein the method further comprises:

for each identified program, determining a number of times that rebooting as a remedial measure is acceptable;

for each identified program, incrementing a counted associated with the identified program each time the system is rebooted due to the failure of that identified program to check-in; and

halting system operation when the number of times the system is rebooted due to the failure of an identified program to check-in exceeds the number determined for that identified program.

14. A method, as set forth in claim 13, wherein the counters are reset based on a user specified condition.

15. A method, as set forth in claim 14, wherein the user specified condition is expiration of a predetermined period of time.

16. An embedded system comprising:

a processor responsive to programs including an operating system;

a watched memory location;

at least one watched program that stores an identifier and a delta time in the watched memory location, the watched program being configured to periodically write a timestamp associated with the identifier in the watched memory, the period being less than or equal to the delta time; and

a watchdog program that periodically checks for failures in the watched programs by comparing the timestamps for each watched program to the difference of a current time and the delta time for that watched program, when a failure is identified the watchdog program executes a remedial action associated with the watched program.

17. An embedded system, as set forth in claim 16, wherein the remedial action is a restart of the system and the watched program is configured to provide an maximum number of failures; and

the watchdog program includes counters that keep track of the number of failures for each watched program and when the number of failures for any watched program is equal to or exceeds that watched programs maximum number, the watchdog program halts the embedded system.

18. An embedded system, as set forth in claim 17, wherein the watchdog program resets the counters upon the expiration of a predetermined time.

19. An embedded system, as set forth in claim 17, wherein the watchdog program resets the counters each day.

20. An embedded system, as set forth in claim 16, wherein a list of remedial actions is defined for each watched program; and

the watchdog program includes counters that keep track of the number of failures for each watched program, the counters being used to select a different remedial action from the list of remedial action to execute upon each failure.

21. An embedded system, as set forth in claim 20, wherein the remedial actions include: restarting the watched program; illuminating an indicator; restarting the system; and halting the operation of the system.

22. An embedded system, as set forth in claim 16, wherein the watched memory location comprises a registry associated with the operating system.

23. An embedded system as set forth in claim 16, wherein the watchdog program is encoded as a service.

24. An embedded system as set forth in claim 16, wherein the watched programs are configured by linking to a common dynamic linked library that contains registration and check-in routines, the registration routines controlling the storing of identifiers and delta times in the watched memory location, and the check-in routines controlling the periodic writing of time stamps.

25. A headless embedded system comprising:

at least one watched program that stores an identifier and a delta time in a watched memory location, the watched program being configured to periodically write a timestamp associated with the identifier in the watched memory, the period being less than or equal to the delta time; and

a watchdog means for periodically checks for failures in the watched programs by comparing the timestamps for each watched program to the difference of a current time and the delta time for that watched program, when a failure is identified the watchdog program executes a remedial action associated with the watched program.